

Vorentwurf  
einer  
Heft-Erweiterung

hier

Lisp-Besprechung  
Meine WM-Tools

Ausrichten nach Z-Werten



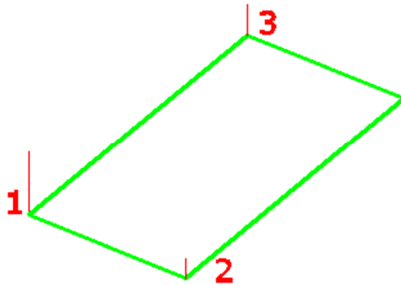
## Z-Ausrichtung

Die Aufgabe ist, eine Platte so im Raum auszurichten, dass 3 der 4 Eckpunkte eine genau definierte Höhe (Z-Wert) aufweisen und keine Verzerrung der Plattengröße entsteht.

## Zeichnerische Lösung

Ich zeichne zunächst die Platte als Rechteck:

- 1) `LAM` (WM-Tool: Layer erstellen)
- 2) `50`
- 3) `3`
- 4)
- 5) `RechtEck`
- 6) `0,0`
- 7) `2500,1250`



*Das grüne Rechteck soll die Platte darstellen. Die roten Striche die Höhenänderung, die Zahlen die Reihenfolge der Punktauswahl.*

*Der Ausgangspunkt ist jeweils an der Platte. Die roten Linie soll nur als optische Hilfe dienen zur Kontrolle der Z-Höhe.*

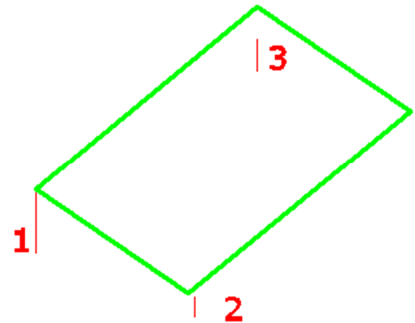
Die Höhe-Änderungen sind bei **1** 500, bei **2** 150 und bei **3** 250 Ein-

heiten.

Richte ich nun die Platte aus, indem ich die einzelnen Punkt entsprechend anklicke:

- 8) `AUSrichten`
- 9) `0,100` (Klick auf die Platte)
- 10)
- 11) `0,1250`
- 12) `0,1250,500`
- 13) `0,0`
- 14) `0,0,150`
- 15)
- 16)

Erhalte ich folgendes Bild:



Zumindest die ersten beiden Punkte scheinen zu stimmen. Den 3. Eck-Punkt beachte ich zunächst nicht. Prüfe ich die Punkte aber z.B. mit dem Befehl `id`, erhalte ich für den 2. Eck-Punkt:

```
Befehl: id
Punkt angeben: end
von X = 0.0 Y = 46.3 Z = 163.0
```

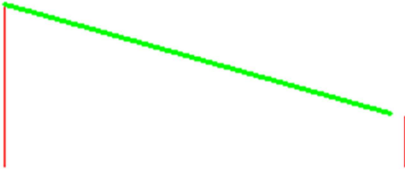
Die neue Höhe sollte aber 150 betragen.

*Es liegt daran, dass die Verlängerung der Linie bei der Höhe 150 auslaufen würde. Ich habe aber eine Kantenlänge von 1250, die*



nur horizontal bis zur Hilfslinie reichen würde.

Von der Seite betrachtet sieht meine Platte so aus:



Die grüne Linie (=Plattenkante) hat eine Länge von 1250 Einheiten und die roten Linien haben einen horizontalen Abstand von 1250 Einheiten. Die linke rote Linie ist 500 hoch, die rechte Linie 150.

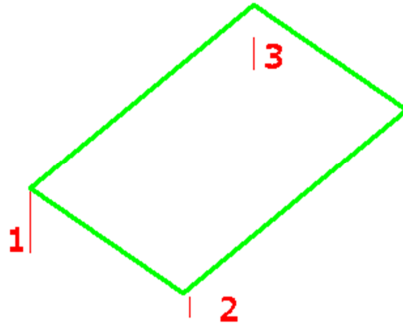
Damit die Höhe 150 an der Platten-Ecke exakt erreicht wird, muss ich ebenfalls den horizontalen (hier Y-) Wert exakt angeben.

Da es sich bei der Konstellation um eine rechtwinklige Anordnung handelt, hilft der Satz von Pythagoras  $c^2=a^2+b^2$ . Wenn  $c$  die Länge der grünen Linie (1250) ist und  $a$  die Höhenänderung (500-150=350), dann ergibt sich als horizontale Länge:  $b = \sqrt{c^2 - a^2} = \sqrt{1250^2 - 350^2} = 1200$ . Der neue Y-Wert ist damit  $1250-1200=50$ !

In AutoCAD wie folgt eingegeben erhalte ich folgendes Bild:

- 17) **AUSrichten**
- 18) 0,100 (Klick auf die Platte)
- 19)
- 20) 0,1250
- 21) 0,1250,500
- 22) 0,0

- 23) 0,50,150
- 24)
- 25)



Die Überprüfung mit ID bestätigt:

```
Befehl: id
Punkt angeben: end
von X = 0.0 Y = 50.0 Z = 150.0
```

Der Z-Wert von Ecke 2 hat die gewünschte Höhe von 150.

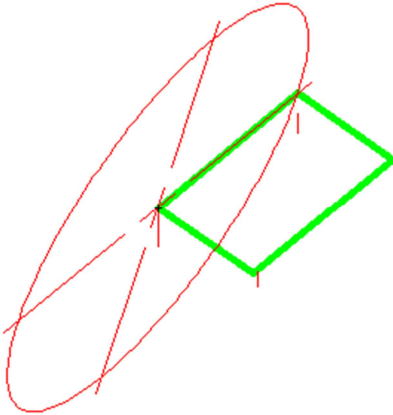
Nun überprüfe ich die Ecke 3, ebenfalls mit dem ID-Befehl:

```
Befehl: ID
Punkt angeben: X = 2500.0 Y = 1250.0 Z = 500.0
```

Hier ist der gewünschte Wert 250 Einheiten groß.

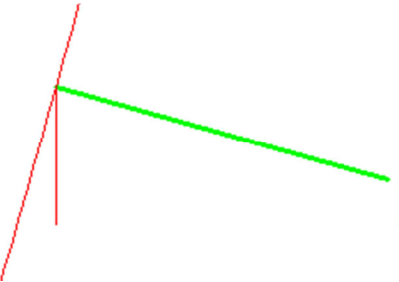
Der 3. Eckpunkt wird ausgerichtet, indem um die zuerst ausgerichtete Kante gedreht wird. Um mir dies bildlich vorzustellen zeichne ich mir einen Kreis um den Drehpunkt. Muss aber vorher die richtige Drehachse einstellen:

- 26) **BKS**
- 27) **ZAchse**
- 28) 0,1250,500
- 29) 0,50,150
- 30) **Kreis**
- 31) 0,0
- 32) 2500



Zur Verdeutlichung habe ich den Kreis hier mit Symmetrieachsen dargestellt.

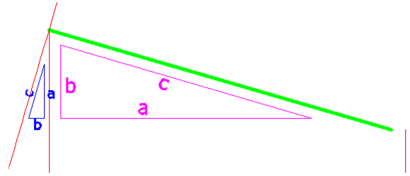
In der Ansicht von links, wird das Dilemma deutlicher:



Die schräge rote Linie gibt einen Teil des gerade gezeichneten Kreises wieder. Und das Dilemma ist, dass ich keinen Schnittpunkt finden kann.

Ich kann aber einen Punkt berechnen, der als Beginn einer weiteren Hilfslinie einen Schnittpunkt mit dem Kreis ermöglicht.

Ich schaue mir das obige Bild noch etwas detaillierter an:



Die Werte des magentafarbenen Dreiecks haben wir bereits ermittelt. Vom blauen Dreieck kennen wir nur  $a$ . Wir wissen aber, dass die Winkel und damit die Seitenverhältnisse gleich sind.

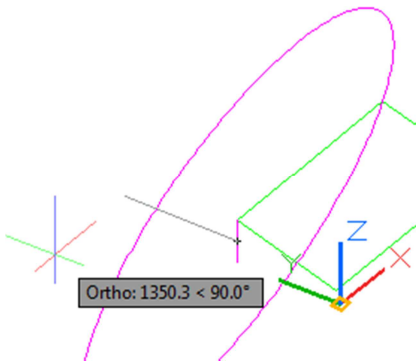
Damit können wir  $b$  vom blauen Dreieck bestimmen und haben einen ersten Punkt für eine weitere Hilfslinie:

$$b = a * \frac{b}{a} = 250 * \frac{350}{1200}$$

Ich habe bewusst kein Ergebnis eingetragen, da im Idealfall die Berechnung von AutoCAD übernommen wird. In AutoCAD zeichne ich eine Hilfslinie, muss dafür wieder zurück in das Welt-BKS:

- 33) BKS
- 34) Welt
- 35) Linie
- 36) 0,1250,250

Jetzt richte ich mich auf eine halb-automatische Punkteingabe ein. D.h. ich schalte den Ortho-Modus ein (F8) und bewegen das Fadenkreuz so, dass die Gummiband-Linie in die gewünschte Richtung (positive Y-Achse) zeigt:

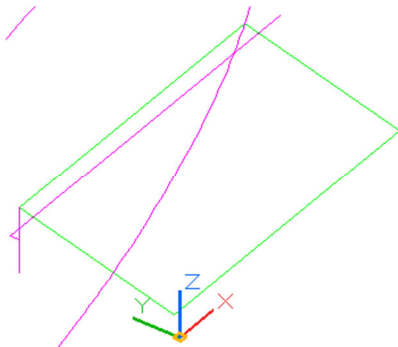


- 44) **DreHen**
- 45) **0,100**
- 46)
- 47) **0,0**
- 48) **Bezug**
- 49) **0**
- 50) **SCH**

Und gebe nun die Formel ein:

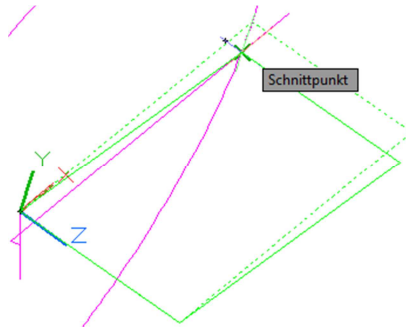
- 37) **(\* (/ 350.0 1200)**
- 38) **250.0)**
- 39) **@3000,0**

*Die Kommastellen (350.0 und 250.0) sind wichtig, damit AutoCAD nicht nur mit Ganzzahlen rechnet und als Ergebnis 0 wiedergibt.*



Ich drehe meine Platte:

- 40) **BKS**
- 41) **ZAchse**
- 42) **0,1250,500**
- 43) **0,50,150**



Nach dem Mausklick an oben dargestelltem Schnittpunkt, wechsele ich wieder in das Welt-Koordinatensystem und prüfe den Eckpunkt 3 meiner Ausrichtung mit dem ID-Befehl:

```
Befehl: ID
Punkt angeben: end
von X = 2486.4 Y = 1322.9 Z = 250.0
```

Damit ist die mir gestellte Aufgabe zeichnerisch mit ein wenig Geometriekenntnissen perfekt gelungen.

Diese Schritte waren noch einigermaßen schnell zu machen. Kommt noch eine Drehung auf der XY-Ebene hinzu, muss ich das BKS entsprechend anpassen.

Habe ich diese Aufgabe häufiger zu machen, lohnt sich vermutlich eine Automatisierung mit LISP.



## Automatisierung

Ich nenne meine Routine ZAUS:

```
(defun c:zaus ()
  (VE)
  (hier steht das Herzstück)
  (Re)
) ;end defun
```

*VE und RE sind Vor- und Rückstellungen, die ich in den WM-Tools definiert habe.*

Im Herzstück frage ich zunächst die Punkte ab. Der Ablauf ist identisch der normalen Abfrage mit dem Befehl AUSRichten, jedoch frage ich auch die Z-Höhen der einzelnen Punkte ab.

Im Herzstück schreibe ich somit:

```
(setq
  Obj (ssget)
  p1a (getpoint "\n Basispunkt = ")
  p1z (getpoint p1a "\n Ziel für
    Basispunkt = ")
  p2a (getpoint "\n 2. Ausgangs-
    punkt = ")
  p2z (getpoint "\n 2. Zielpunkt für
    Richtung auf BKS-Ebene = ")
  h2 (getreal "\n Höhe für 2. Ziel-
    punkt =")
  p3a (getpoint "\n 3. Ausgangs-
    punkt = ")
  h3 (getreal "\n Höhe für 3. Ziel-
    punkt ="))
```

*Der 2. Zielpunkt soll die Richtung auf der XY-Ebene definieren, um dann die Höhe für diesen Punkt zu bekommen. Vom 3. Punkt benötige ich nur den Ausgangspunkt zur Bestimmung der Plattengröße und die Ziel-Höhe, da ich davon ausgehe, dass die Platte rechtwinklig*

*ist.*

Jetzt ermittle ich die Plattengröße:

```
(setq Dist1 (distance p1a p2a)
  Dist2 (distance p1a p3a))
```

Um die neue Richtung der ersten Kante zu bestimmen:

```
(setq Win1
  (angle (2DP p1z) (2DP p2z)))
```

*2DP ist eine kleine Routine aus den WM-Tools, die mir nur die YX-Werte eines Punktes ausliest. Also*

```
(2DP p1z) steht für
(list (car p1z)(cadr p1z))
```

Um damit den neuen horizontalen Abstand - mit den Satz von Pythagoras - zu ermitteln:

$$b = \sqrt{c^2 - a^2}$$

In Lisp sieht die Formel so aus:

```
(setq b (sqrt (- (expt c 2)(expt a 2))))
```

*Sqrt steht für Wurzel, expt c 2 steht für c<sup>2</sup>.*

In meinem Beispiel erzeuge ich für diese Formel eine kleine Unter-Routine. Ich nenne sie ‚Kathede‘:

```
(defun Kathede ( c a)
  (sqrt (- (expt c 2) (expt a 2)))
)
```

In meiner Routine ZAUS setze ich Dist1a für b. Dist1 steht für c und für die Differenz der Soll-Z-Werte von Punk 1 und 2.

```
(setq Dist1a
  (Kathede
    dist1 (- (caddr p1z) h2)))
```

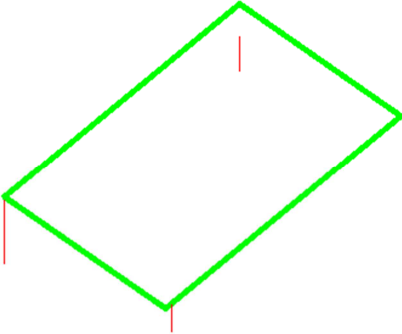
Und definiere den Ziel-Punkt 2:

```
(setq p2 (polar p1z Win1 Dist1a))
(setq p2 (list (car p2)(cadr p2) h2))
```



Nun kann ich den ersten Schritt schon mal probieren:

Hurra, es hat geklappt. Auch der ID-Wert des 2. Eckpunktes hat die gewünschte Höhe von 150.



Dann weiter an die Drehung um erste Kante. Ich setze die Z-Achse des BKS auf die erste Kante, Ursprung ist der Ziel-Punkt 1:

(command „BKS“ „ZA“ p1z p2)